IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: Gilbert Wolrich et al.	Art Unit	: 2183
Serial No.	: 10/069,352	Examiner	: David J. Huisman
Filed	: August 7, 2002	Conf. No.	: 7931
Title	: FAST WRITE INSTRUCTION FOR MICRO ENGINE USED IN MULTITHREADED PARALLEL PROCESSOR ARCHITECTURE		

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

BRIEF ON APPEAL

Please charge the brief fee of $510 and please apply any other charges or credits to Deposit Account No. 06-1050.

CERTIFICATE OF MAILING BY EFS-WEB FILING

I hereby certify that this paper was filed with the Patent and Trademark Office using the EFS-WEB system on this date: March 10, 2008

(1) Real Party in Interest

The real party in interest in the above application is Intel Corporation.

(2) Related Appeals and Interferences

The Appellant is not aware of any appeals or interferences related to the above-identified patent application.

(3) Status of Claims

This is an appeal from the decision of the Primary Examiner in a Final Office Action dated October 11, 2007. Claims 1-38 are pending in the application. Of these, claims 1-6, 8-11, 14, 17, 20-25, 27-30, 33, and 36 were rejected. Claims 7, 12, 13, 15, 16, 18, 19, 26, 31, 32, 34, 35, 37, and 38 were objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Claims 1-6, 8-11, 14, 17, 20-25, 27-30, 33, and 36 are the subject of this appeal.

(4) Status of Amendments

Appellant filed a Reply to the Non-Final Office Action of May 3, 2007, amending claims 1, 6, 20, and 25. Appellant filed a Reply to the Final Action of October 11, 2007. No amendments from the Reply to the Final Office Action have been entered. Appellant filed a Notice of Appeal on **January 9, 2008**.

(5) Summary of Claimed Subject Matter

Claim 1

One aspect of Appellant's invention is set out in claim 1 as a method of operating a multi-threaded processor. *"...a communication system 10 includes a parallel, hardware-based multithreaded processor 12."*[1]

[1] Specification, page 2, lines 2-3.

Inventive features of claim 1 include receiving data specified by execution of a fast-write instruction in one of multiple threads processing on the multi-threaded processor, the one of the multiple threads identified by a processing thread number, the fast-write instruction further specifying a register, the register having multiple groups of bits, each group of bits associated with a corresponding thread of the multiple threads processing on the multi-threaded processor. *"The computer instruction architecture set ... includes a fast write (FAST_WR) instruction. The FAST_WR instruction for immediate data writes specified immediate data to a specified control and status register (CSR) by having write data specified directly in the instruction rather than in a transfer register."*[2] *"The FBI unit automatically shifts the immediate data into the appropriate register field corresponding to the thread that is writing the FAST_WR data."*[3] *"The 'inter_thd_sig' field represents a thread number (0 - 23) of the thread to be signaled."*[4]

Inventive features of claim 1 also include selecting a group of bits associated with the one of the multiple threads, the group of bits being selected from the multiple groups of bits of the register specified by the fast-write instruction according to the processing thread number. *"A format for the fast write instruction is: fast_wr [immed_data, csr_addr], optional_token, ... The "immed_data" field represents 10 bits of the immediate data to be written to the control and status register (CSR); valid immed_data values are 0 through 0x3FF. The "csr_addr" field represents the symbolic names that follow address the corresponding CSRs."*[5]

Inventive features of claim 1 also include loading the data into the bit positions of the selected group of bits of the register. *"FAST_WR writes the specified immediate data to the specified FBI [FIFO Bus Interface] CSR."*[6] *"The 10-bit immediate data supplied with the instruction is shifted in two segments to the appropriate fields. Bits 6 through 0 are shifted left by an amount equal to the thread number writing the data. Bits 9 through 7 are always shifted into the BP2 through BP0 positions regardless of the micro engine writing the data."*[7]

---

[2] Id., Page 10, Lines 17-20.
[3] Id., Page 10, Lines 22-24.
[4] Id., Page 11, Lines 4-5.
[5] Id., Page 10, Lines 25-30.
[6] Id., Page 10, Lines 20-21.
[7] Id., Page 11, Lines 6-9.

Claim 20

Another aspect of Appellant's invention is set out in claim 20 as a computer program product, disposed on a computer readable medium comprising a fast-write instruction. *"The hardware-based multithreaded processor 12 ... includes a central controller 20 that assists in loading micro code control for other resources of the hardware-based multithreaded processor 12 and performs other general purpose computer type functions..."*[8] *"The general-purpose microprocessor 20 has an operating system. Through the operating system the processor 20 can call functions to operate on micro engines 22a-22f."*[9] *"The micro engines 22 support an instruction set."*[10] *"The computer instruction architecture set also includes a fast write (FAST_WR) instruction."*[11]

Inventive features of claim 20 include the fast-write instruction causing a computer to receive data specified in the fast-write instruction in one of multiple threads processing on a multi-threaded processor, the one of the multiple threads identified by a processing thread number, the fast-write instruction further specifying a register, the register having multiple groups of bits, each group of bits associated with a corresponding thread of multiple threads processing on the multi-threaded processor. This feature finds support as the analogous feature of claim 1.

Inventive features of claim 20 also include the fast-write instruction causing a computer to select a group of bits associated with the one of the multiple threads, the group of bits being selected from the multiple groups of bits of the register specified in the fast-write instruction according to the processing thread number. This feature finds support as the analogous feature of claim 1.

Inventive features of claim 20 also include the fast-write instruction causing a computer to load the data into the bit positions of the selected group of bits of the register. This feature finds support as the analogous feature of claim 1.

---

[8] Specification, Page 2, Lines 10-12.
[9] Id., Page 2, Lines 16-18.
[10] Id., Page 10, Line 13.
[11] Id., Page 10, Lines 17-18.

(6) Grounds of Rejection to be Reviewed on Appeal

(1) Claims 1, 2, 6, 8, 9, 20, 21, 25, 27, and 28 stand rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,704,054 ("Bhattacharya").

(2) Claims 3-5, 10, 11, 14, 17, 22-24, 29, 30, 33, and 36 rejected under 35 U.S.C. § 103(a) as being unpatentable over Bhattacharya.

(7) Argument

Anticipation

"It is well settled that anticipation under 35 U.S.C. §102 requires the presence in a single reference of all of the elements of a claimed invention." *Ex parte Chopra*, 229 U.S.P.Q. 230, 231 (BPA&I 1985) and cases cited.

"Anticipation requires the presence in a single prior art disclosure of all elements of a claimed invention arranged as in the claim." *Connell v. Sears, Roebuck & Co.*, 220 U.S.P.Q. 193, 198 (Fed. Cir. 1983).

"This court has repeatedly stated that the defense of lack of novelty (i.e., 'anticipation') can only be established by a single prior art reference which discloses each and every element of the claimed invention." *Structural Rubber Prod. Co. v. Park Rubber Co.*, 223 U.S.P.Q. 1264, 1270 (Fed. Cir. 1984), citing five prior Federal Circuit decisions since 1983 including *Connell*.

In a later analogous case the Court of Appeals for the Federal Circuit again applied this rule in reversing a denial of a motion for judgment n.o.v. after a jury finding that claims were anticipated. *Jamesbury Corp. v. Litton Industrial Prod., Inc.*, 225 U.S.P.Q. 253 (Fed. Cir. 1985).

After quoting from *Connell*, "Anticipation requires the presence in a single prior art disclosure of all elements of a claimed invention arranged as in the claim," 225 U.S.P.Q. at 256, the court observed that the patentee accomplished a constant tight contact in a ball valve by a lip on the seal or ring which interferes with the placement of the ball. The lip protruded into the area where the ball will be placed and was thus deflected after the ball was assembled into the valve. Because of this constant pressure, the patented valve was described as providing a particularly good seal when regulating a low pressure stream. The court quoted with approval

from a 1967 Court of Claims decision adopting the opinion of then Commissioner and later Judge Donald E. Lane:

> [T]he term "engaging the ball" recited in claims 7 and 8 means that the lip contacts the ball with sufficient force to provide a fluid tight seal **** The Saunders flange or lip only sealingly engages the ball 1 on the upstream side when the fluid pressure forces the lip against the ball and never sealingly engages the ball on the downstream side because there is no fluid pressure there to force the lip against the ball. The Saunders sealing ring provides a compression type of seal which depends upon the ball pressing into the material of the ring. *** The seal of Saunders depends primarily on the contact between the ball and the body of the sealing ring, and the flange or lip sealingly contacts the ball on the upstream side when the fluid pressure increases. 225 U.S.P.Q. at 258.

Relying on *Jamesbury*, the ITC said, "Anticipation requires looking at a reference, and comparing the disclosure of the reference with the claims of the patent in suit. A claimed device is anticipated if a single prior art reference discloses all the elements of the claimed invention as arranged in the claim." *In re Certain Floppy Disk Drives and Components Thereof*, 227 U.S.P.Q. 982, 985 (U.S. ITC 1985).

Obviousness

"It is well established that the burden is on the PTO to establish a prima facie showing of obviousness, *In re Fritsch*, 972 F.2d. 1260, 23 U.S.P.Q.2d 1780 (C.C.P.A., 1972)."

In *KSR International Co. v. Teleflex Inc.*, 550 U.S. __, (2007), the Supreme Court reversed a decision by the Court of Appeal's for the Federal Circuit decision that reversed a summary judgment of obviousness on the ground that the district court had not adequately identified a motivation to combine two prior art references. The invention was a combination of a prior art repositionable gas pedal, with prior art electronic (rather than mechanical cable) gas pedal position sensing. The Court first rejected the "rigid" teaching suggestion motivation (TSM) requirement applied by the Federal Circuit, since the Court's obviousness decisions had

all advocated a "flexible" and "functional" approach that cautioned against "granting a patent based on the combination of elements found in the prior art."

With respect to the genesis of the TSM requirement, the Court noted that although "As is clear from cases such as Adams[12], a patent composed of several elements is not proved obvious merely by demonstrating that each of its elements was, independently, known in the prior art. Although common sense directs one to look with care at a patent application that claims as innovation the combination of two known devices according to their established functions, it can be important to identify a reason that would have prompted a person of ordinary skill in the relevant field to combine the elements in the way the claimed new invention does. This is so because inventions in most, if not all, instances rely upon building blocks long since uncovered, and claimed discoveries almost of necessity will be combinations of what, in some sense, is already known."

In application of the TSM requirement, the Court cautioned that: "Helpful insights, however, need not become rigid and mandatory formulas; and when it is so applied, the TSM test is incompatible with our precedents."

"The mere fact that the prior art could be so modified would not have made the modification obvious unless the prior art suggested the desirability of the modification." *In re Gordon*, 221 U.S.P.Q. 1125, 1127 (Fed. Cir. 1984).

> **Although the Commissioner suggests that [the structure in the primary prior art reference] could readily be modified to form the [claimed] structure, "[t]he mere fact that the prior art could be so modified would not have made the modification obvious unless the prior art suggested the desirability of the modification." *In re Laskowski*, 10 U.S.P.Q. 2d 1397, 1398 (Fed. Cir. 1989).**

"The claimed invention must be considered as a whole, and the question is whether there is something in the prior art as a whole to suggest the desirability, and thus the obviousness, of making the combination." *Lindemann Maschinenfabrik GMBH v. American Hoist & Derrick*, 221 U.S.P.Q. 481, 488 (Fed. Cir. 1984).

---

[12] United States v. Adams, 383 U. S. 39, 40 (1966)

> **Obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention, absent some teaching or suggestion supporting the combination. Under Section 103, teachings of references can be combined only if there is some suggestion or incentive to do so. *ACS Hospital Systems, Inc. v. Montefiore Hospital*, 221 U.S.P.Q. 929, 933 (Fed. Cir. 1984) (emphasis in original, footnotes omitted).**

"The critical inquiry is whether 'there is something in the prior art as a whole to suggest the desirability, and thus the obviousness, of making the combination.'" *Fromson v. Advance Offset Plate, Inc.*, 225 U.S.P.Q. 26, 31 (Fed. Cir. 1985).

(1) Claims 1, 2, 6, 8, 9, 20, 21, 25, 27, and 28 are not anticipated by Bhattacharya.

Claim 1

For the purposes of this appeal only, claims 1, 8, 9, 20, 27, and 28 stand or fall together. Claim 1 is representative of this group of claims.

Claim 1 is directed to a method of operating a multithreaded parallel processor. Appellant contends that Bhattacharya neither describes nor suggests at least the features of "selecting a group of bits associated with the one of the multiple threads, the group of bits being selected from...multiple groups of bits of [a] register specified by [a] fast-write instruction according to [a] processing thread number," as called for in claim 1.

The examiner contends that for

> **...claim 1, Bhattacharya has taught a method of operating a multi-threaded processor comprising:**
>
> **a) receiving data specified by execution of a fast-write instruction in one of multiple threads processing on the multi-threaded processor, the one of the multiple threads identified by a processing thread number. See column 6, lines 32-57. Threads, which are inherently identified by number, and at least also by address in Bhattacharya, also include instructions which specify result data.**
>
> **b) the fast-write instruction further specifying a register, the register having multiple groups of bits, each group of bits associated with a corresponding thread of the multiple threads processing on the multi-threaded processor. See Fig.6, component 42, and column 6, lines 32-41. Note that a result register, which holds results specified by a "fast-write" instruction, is divided into at least N groups for N threads. Each thread will then write to its portion of the register based on an address identifier.**

> **c) selecting a group of bits associated with the one of the multiple threads, the group of bits being selected from the multiple groups of bits of the register specified by the fast-write instruction according to the processing thread number. Again, see column 6, lines 32-57. When a thread is to write to the result register specified by the fast write, the thread number (address identifier) is used to select the group of bits (portion) to which the result is written.**
>
> **d) loading the data into the selected bit positions of the register. See column 6, lines 32-57. A thread writes a result to the result register (loads data into the result register).**[13]

Appellant disagrees. Bhattacharya, in the places referenced by the examiner, teaches the following:

> **In FIG. 6 an exemplary pipeline stage is shown that is employed in the multi-thread pipeline structure of FIG. 5. Each source and destination latch in instruction register 40 is provided with plural address positions 106. Result latches 42 are similarly provided with plural address positions 108. Assuming that there are three threads being executed on a time-shared basis, each instruction register 40 and result register 42 must contain at least three separate address positions to hold source and destination/result values from the respective threads.**
>
> **Sequencer modules 110 and 112 respectively provide address designations in succeeding pipeline stages for both the result values flowing in the result pipeline and the instructions flowing in the instruction pipeline. For instance, sequencer 110 will provide output addresses in a next pipeline stage to which result values residing in result register 42 will be directed. Similarly, sequencer 112 provides an input address for results coming from a prior pipeline stage. Sequencer 110 provides an input address for instructions entering instruction register 40 and sequencer 122 provides output addresses to which the instruction in register 40 will be directed in a subsequent pipeline stage.**
>
> **Assuming, as indicated above, that there are three threads in process, sequencers 110 and 112 sequence in a round-robin manner through the three threads and cause movement of individual instructions/results of each thread.** [14]

Bhattacharya neither describes nor suggests in these passages the feature of: "...the group of bits being selected from...multiple groups of bits of [a] register specified by [a] fast-write instruction ...". Rather, in the relied on passages Bhattacharya explains that within a pipeline an instruction register and a results register are each provided with plural address positions, and that

---

[13] Final Office Action dated October 11, 2007, Paragraph 9, Pages 3-4.
[14] Bhattacharya, Column 6, Lines 32-57, and FIG. 6

each register must contain at least enough registers to store instructions and data associated with each thread.

The relationship between an instruction and data is disclosed by Bhattacharya as follows:

> **Each of the source and destination registers includes a section 58 that carries a binding value (e.g. a register name in register file 12). In similar manner, result registers 60 and 62…contain binding value sections 64. By comparison of binding values 58 and 64 in logic circuit 44, a pipeline stage is able to determine which data passing through result register 42 is to be associated with an instruction in instruction register 40.**[15]

That is, in Bhattacharya, the instructions in the source register and the data in the results register are mapped by binding values in a logic circuit. However, this is not what is called for in claim 1, which requires that "the fast-write instruction further specifying a register," as required by Appellant's independent claim 1. Because Bhattacharya neither discloses nor suggests at least the features of "selecting a group of bits associated with the one of the multiple threads, the group of bits being selected from...multiple groups of bits of [a] register specified by [a] fast-write instruction according to [a] processing thread number", Appellant's independent claim 1 is not anticipated by Bhattacharya.

Accordingly because Bhattacharya does not describe of all elements of the claims arranged as in the claims, the claims cannot be anticipated by Bhattacharya.

Claim 2

Claim 2 is directed to a method of operating a multithreaded parallel processor and depends from claim 1. Bhattacharya neither describes nor suggests at least the feature of "the register [being] a control and status register (CSR)."

The examiner contends that for

> **…claim 2, Bhattacharya has taught a method as described in claim 1. Bhattacharya has further taught that the register is a control and status register (CSR). See Fig.6, component 42, and note that a result register, when used as an operand of a dependent instruction, controls that**

[15] Id., Column 3, Lines 6-13

> **instruction. In addition, it is a status register because writing a result implies completion status of the writing instruction.[16]**

Further, the examiner contends that Bhattacharya's results register is a "**control and status register**." Appellant disagrees and contends that Bhattacharya's instruction would then be required to specify the register. Bhattacharya discloses that:

> **[a] pipeline stage wherein an instruction operation code is to be executed is, for instance, adder launch module 32 wherein an add instruction is both recognized and causes data having a proper binding value to be fed to adder 28 wherein a sum is created and then fed to add recover module 34. The result data then propagates to register file 12 and is stored. [17]**

The instruction here merely causes data to be acted upon by some module before it is delivered to some register. Therefore, assuming *arguendo* that this register is a control and status register,[18] claim 2 still distinguishes over Bhattacharya because the register, as disclosed by Bhattacharya is not specified by the instruction, as required by Appellant's claim 2. Therefore, Bhattacharya neither describes nor suggests at least the features of "…the fast-write instruction … specifying a register…", and thus Appellant's claim 2 is not anticipated by Bhattacharya.

Claim 6

Claim 6 is directed to a method of operating a multithreaded parallel processor and depends from claim 1. Bhattacharya neither describes nor suggests at least the features of "the one of multiple threads is processed on a micro engine of a multi-threaded processor".

The examiner contends that in

> **…claim 6, Bhattacharya has taught a method as described in claim 1. Bhattacharya has further taught that the processing thread represents processing in a micro engine of a multithreaded processor. See Fig.5, and note a pipeline (micro engine) processes a thread.[19]**

---

[16] Final Office Action dated October 11, 2007, Paragraph 10, Page 4.
[17] Bhattacharya, Column 2, Lines 54-60, and FIG. 1
[18] Appellant does not concede that this is the case.
[19] Final Office Action dated October 11, 2007, Paragraph 11, Pages 4-5.

Claim 6 requires that the multi-threaded processor is a parallel, hardware-based multi-threaded processor comprising a plurality of micro engines. These features are neither described nor suggested by Bhattacharya. Rather, Bhattacharya recites the following:

> **In FIG. 5, ... [b]etween each pair of successive pipeline stages (e.g. 100 and 102), there resides a sequencer module 104 which controls movement of instructions and results between pipeline stages.** [20]

Bhattacharya further states:

> **[s]equencer modules ... provide address designations in succeeding pipeline stages for both the result values flowing in the result pipeline and the instructions flowing in the instruction pipeline.** [21]

Bhattacharya provides micro sequencers and not micro engines as required by Appellant's claim 6. Because Bhattacharya neither discloses nor suggests at least the features of "the one of multiple threads is processed on a micro engine of a multi-threaded processor", Appellant's claim 6 is not anticipated by Bhattacharya.

Claims 8 and 9 depend from independent claim 1 and are therefore not anticipated by Bhattacharya for at least the same reasons as independent claim 1.

Independent claim 20 recites "[a] computer program product, disposed on a computer readable medium, the program comprising a fast-write instruction that causes a computer to...select a group of bits associated with the one of the multiple threads, the group of bits being selected from the multiple groups of bits of the register specified in the fast-write instruction according to the processing thread number..." For reasons similar to those provided with respect to independent claim 1, at least these features are not disclosed by Bhattacharya. Appellant's independent claim 20 is therefore not anticipated by Bhattacharya.

Claims 21, 25, 27 and 28 depend from independent claim 20, are analogous to claims 2, 6, 8 and 9, respectively. Therefore Appellant's claims 21, 25, 27 and 28 are not anticipated by Bhattacharya for at least the same reasons as claims 2, 6, and independent claim 20.

---

[20] Bhattacharya, Column 6, Lines 32
[21] Id., Column 6, Lines 42-45

(2) Claims 3-5, 10, 11, 14, 17, 22-24, 29, 30, 33, and 36 are not unpatentable over Bhattacharya.

For the purposes of this appeal only, claims 3-5, 10, 11, 14, 17, 22-24, 29, 30, 33, and 36 stand or fall together.

Claims 3-5, 10, 11, 14, and 17 depend from claim 1, and claims 22-24, 29, 30, 33, and 36 depend from claim 20. Because of the above statements showing that claims 1 and 20 are not anticipated by Bhattacharya, Appellant's claims 3-5, 10, 11, 14, 17, 22-24, 29, 30, 33, and 36 are not unpatentable over Bhattacharya.

Appellant submits, therefore, that claims 1-6, 8-11, 14, 17, 20-25, 27-30, 33, and 36 are neither anticipated by nor unpatentable and are otherwise allowable over Bhattacharya. Therefore, the Examiner erred in rejecting Appellant's claims and should be reversed.

Respectfully submitted,

Date: March 10, 2008

/Denis G. Maloney/
Denis G. Maloney
Reg. No. 29,670

Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

Appendix of Claims

1. A method of operating a multi-threaded processor comprising:

receiving data specified by execution of a fast-write instruction in one of multiple threads processing on the multi-threaded processor, the one of the multiple threads identified by a processing thread number, the fast-write instruction further specifying a register, the register having multiple groups of bits, each group of bits associated with a corresponding thread of the multiple threads processing on the multi-threaded processor;

selecting a group of bits associated with the one of the multiple threads, the group of bits being selected from the multiple groups of bits of the register specified by the fast-write instruction according to the processing thread number; and

loading the data into the bit positions of the selected group of bits of the register.

2. The method of claim 1 wherein the register is a control and status register (CSR).

3. The method of claim 2 wherein the control and status register is coupled to a 64-bit wide first-in first-out (FIFO) bus.

4. The method of claim 3 wherein the FIFO bus interfaces with Media Access Controller (MAC) devices.

5. The method of claim 1 wherein the data represents hexadecimal mask values 0 to 0x3FF.

6. The method of claim 1 wherein the one of multiple threads is processed on a micro engine of a multi-threaded processor.

Claim 7 was objected to as being dependent on a rejected base claim.

8. The method of claim 1 wherein the fast-write instruction comprises a token.

9. The method of claim 8 wherein the token represents overriding qualifiers.

10. The method of claim 8 wherein the token is a 32-bit word.

11. The method of claim 10 wherein a token format comprises:
an OV field in bit 31;
a micro engine (UENG) ADDR field in bits 30:28;
a reserved field in bits 27:16;
an OV field in bit 15;
a fast write data field in bits 14:5;
a reserved field in bits 4:3;
an OV field in bit 2; and
a CTX field in bits 1:0.

Claims 12 and 13 were objected to as being dependent on a rejected base claim.

14. The method of claim 11 wherein bits 27:16 return 0 when read.

Claims 15 and 16 were objected to as being dependent on a rejected base claim.

17. The method of claim 11 wherein bits 4:3 return 0 when read.

Claims 18 and 19 were objected to as being dependent on a rejected base claim.

20. A computer program product, disposed on a computer readable medium, the program comprising a fast-write instruction that causes a computer to:

receive data specified in the fast-write instruction in one of multiple threads processing on a multi-threaded processor, the one of the multiple threads identified by a processing thread number, the fast-write instruction further specifying a register, the register having multiple

groups of bits, each group of bits associated with a corresponding thread of multiple threads processing on the multi-threaded processor;

select a group of bits associated with the one of the multiple threads, the group of bits being selected from the multiple groups of bits of the register specified in the fast-write instruction according to the processing thread number; and

load the data into the bit positions of the selected group of bits of the register.

21. The computer program product of claim 20 wherein the register is a control and status register (CSR).

22. The computer program product of claim 21 wherein the control and status register is coupled to a 64-bit wide first-in first-out (FIFO) bus.

23. The computer program product of claim 22 wherein the FIFO bus interfaces with Media Access Controller (MAC) devices.

24. The computer program product of claim 20 wherein the data represents hexadecimal mask values 0 to 0x3FF.

25. The computer program product of claim 20 wherein the one of the multiple threads is processed on a micro engine of a multi-threaded processor.

Claim 26 was objected to as being dependent on a rejected base claim.

27. The computer program product of claim 20, wherein the fast-write instruction comprises a token.

28. The computer program product of claim 27 wherein the token represents overriding qualifiers.

29. The computer program product of claim 27 wherein the token is a 32-bit word.

30. The computer program product of claim 29 wherein a token format comprises:

an OV field in bit 31;

a micro engine (UENG) ADDR field in bits 30:28;

a reserved field in bits 27:16;

an OV field in bit 15;

a fast write data field in bits 14:5;

a reserved field in bits 4:3;

an OV field in bit 2; and

a CTX field in bits 1:0.

Claims 31 and 32 were objected to as being dependent on a rejected base claim.

33. The computer program product of claim 30 wherein bits 27:16 return 0 when read.

Claims 34 and 35 were objected to as being dependent on a rejected base claim.

36. The computer program product of claim 30 wherein bits 4:3 return 0 when read.

Claims 37 and 38 were objected to as being dependent on a rejected base claim.

Evidence Appendix

None

Related Proceedings Appendix

None